Amendments to the Claims:

Following is a complete listing of the claims pending in the application, as amended:

[c1] 1.   (Currently Amended) A method in a computer system for executing code during a system management mode interrupt, the method comprising:

upon occurrence of a system management mode interrupt,

saving state of the computer system;

switching the computer system to protected mode;

executing 32-bit code that uses a global descriptor table that is different from the global descriptor table in use when the system management mode interrupt occurred; and

upon completion of the execution of the 32-bit code,

restoring the saved state of the computer system; and

returning from the occurrence of the interrupt.

[c2] 2.   (Currently Amended) The method of claim 1 wherein the 32-bit code is an operating system kernel for loading and running programs during the occurrence of the system management mode interrupt.

[c3] 3.   (Currently Amended) The method of claim 2 wherein the programs are Windows NT Portable Executable programs.

[c4] 4.   (Currently Amended) The method of claim 1 wherein the computer system is based on an Intel Pentium processor.

5.   (New) A method in a computer system for executing code during a system management mode interrupt, the method comprising:

upon occurrence of a system management mode interrupt,

switching the computer system from the system management mode to another mode; and

executing code that uses a global descriptor table that is different from the global descriptor table in use when the system management mode interrupt occurred.

6. (New) The method of claim 5 including upon completion of the execution of the code, returning from the occurrence of the system management mode interrupt.

7. (New) The method of claim 5 including:

saving state of the computer system; and

upon completion of the execution of the code,

restoring the saved state of the computer system; and

returning from the occurrence of the system management mode interrupt.

8. (New) The method of claim 5 wherein the executing executes 32-bit code.

9. (New) The method of claim 5 wherein the code is an operating system kernel for loading and running programs during the occurrence of the system management mode interrupt.

10. (New) The method of claim 9 wherein the programs are portable executables.

11. (New) The method of claim 5 wherein the computer system is based on an Intel processor.

12. (New) The method of claim 5 wherein the computer system is based on an Intel-compatible processor.

13. (New) The method of claim 5 wherein the programs are selected from the group consisting of a remote console program, a remote boot program, a remote diagnostics program, a remote restart program, and a debugging program.

14. (New) The method of claim 5 wherein the other mode is protected mode.

15. (New) The method of claim 5 wherein the code executes transparently to the foreground operating system.

16. (New) The method of claim 5 wherein the code executes even if the foreground operating system has crashed or stopped.

17. (New) The method of claim 5 wherein the code executes when the foreground operating system crashes or stops.

18. (New) A computer-readable medium containing instructions for a system management mode interrupt routine that allows execution of code above a 1MB boundary, by a method comprising:

saving state of the processor;

switching the processor to protected mode; and

before returning from the interrupt, executing code that is stored above the 1MB boundary.

19. (New) The computer-readable medium of claim 18 including after executing the code, restoring the saved state of the processor and returning from the interrupt.

20. (New) The computer-readable medium of claim 18 wherein the code is executed using a global descriptor table that is different from the global descriptor table in use when the interrupt occurred.

21. (New) The computer-readable medium of claim 18 wherein the processor is a Pentium-based processor.

22. (New) The computer-readable medium of claim 18 wherein the executed code is 32-bit flat address space code.

23. (New) The computer-readable medium of claim 18 wherein the instructions are loaded into system management memory by a BIOS.

24. (New) The computer-readable medium of claim 18 wherein the code is loaded into memory from a ROM.

25. (New) The computer-readable medium of claim 18 wherein the code is loaded into memory from a flash ROM.